PathFindOnPath

The destination string buffer must be long enough to hold the return file path

Sean Barnum, Cigital, Inc. [vita[1]]

2007-04-02

Part "Original Cigital Coding Rule in XML"

Mime-type: text/xml, size: 4985 bytes

Attack Category	• Malicious Input • Path spoofing or confusion problem
Vulnerability Category	• Buffer Overflow • Unconditional
Software Context	• File Path Management
Location	• shlwapi.h
Description	The destination string buffer for the PathFindOnPath() function must be long enough to hold the return file path. The PathFindOnPath() routine searches the PATH (and optionally other path directories) for the file specified. If found, it modifies in place the file variable to be a fully specified path. Because the returned value is a fully specified path, it will typically be longer than the input file name. Therefore, ensure that the destination buffer is at least MAX_PATH characters in length.
APIs	**Function Name** / **Comments** PathFindOnPath PathFindOnPathA — ASCII implementation PathFindOnPathW — Unicode implementation PathFind
Method of Attack	The attacker could overflow the destination buffer by providing a really long filename, by providing long directory names in the "other dirs" parameter, or manipulate the PATH to have a very long directory name where the file is present. In order for any of these to succeed, the attacker would have to choose a directory name where the file in question actually exists. This could severely limit the attackers ability to actually control behavior of the BO, but he could (in theory) make the program crash. In order for any of this to work,

1. http://buildsecurityin.us-cert.gov/bsi/about_us/authors/35-BSI.html (Barnum, Sean)

	the attacker would need to have control of the file system and the ability to create directories and possibly put files in those directories.
Exception Criteria	
Solutions	<table><tr><th>Solution Applicability</th><th>Solution Description</th><th>Solution Efficacy</th></tr><tr><td>Whenever PathFindOnPath() is used.</td><td>The first parameter, pszFile, must be at least MAX_PATH characters in length.</td><td></td></tr></table>
Signature Details	BOOL PathFindOnPath(LPTSTR pszFile, LPCTSTR *ppszOtherDirs);

Examples of Incorrect Code	`TCHAR file[] = TEXT("MyFile.txt"); // Note: Buffer is too small to hold result`
	`LPTSTR pszFile = file;`
	`LPCTSTR otherDirs[] = {TEXT("C:\ \dir1"), TEXT("C:\\dir2"), NULL};`
	`LPCTSTR *ppszOtherDirs = otherDirs;`
	`Bool result = PathFindOnPath(pszFile, otherDirs);`

Examples of Corrected Code	`TCHAR file[MAX_PATH] = TEXT("MyFile.txt"); // Note: Buffer is correctly sized`
	`LPTSTR pszFile = file;`
	`LPCTSTR otherDirs[] = {TEXT("C:\ \dir1"), TEXT("C:\\dir2"), NULL};`
	`LPCTSTR *ppszOtherDirs = otherDirs;`
	`Bool result = PathFindOnPath(pszFile, otherDirs);`

Source Reference	• http://msdn.microsoft.com/library/default.asp?url=/library/en-us/shellcc/platform/shell/reference/shlwapi/path/pathfindonpath.asp[2]
Recommended Resource	
Discriminant Set	<table><tr><td>**Operating System**</td><td>• Windows</td></tr><tr><td>**Languages**</td><td>• C • C++</td></tr></table>

Cigital, Inc. Copyright

The Build Security In (BSI) portal is sponsored by the U.S. Department of Homeland Security (DHS), National Cyber Security Division. The Software Engineering Institute (SEI) develops and operates BSI. DHS funding supports the publishing of all site content.

1. mailto:copyright@cigital.com